Тема: **Разработка и усовершенствование функционала среды программирования OWEN Logic**

**СОДЕРЖАНИЕ**

[Введение 2](#_Toc183186190)

[1 Подходы к построению информационных систем 4](#_Toc183186191)

[1.1 Принципы DDD 4](#_Toc183186192)

[1.2 Структура и компоненты на основе DDD 5](#_Toc183186193)

[1.3 Преимущества применения DDD 6](#_Toc183186193)

[1.4 Применение DDD для расширяемости 6](#_Toc183186193)

[2 Сравнительный анализ с системами-аналогами 7](#_Toc183186196)

[2.1 Codesys 7](#_Toc183186197)

[2.2 TIA Portal (Siemens) 7](#_Toc183186198)

[2.3 Trace Mode 8](#_Toc183186199)

Введение

Современная промышленность требует эффективных инструментов для автоматизации управления и мониторинга технологических процессов. Программируемые реле (ПР) и панели оператора стали незаменимыми компонентами автоматизированных систем благодаря своей гибкости, надежности и экономической эффективности. Для работы с таким оборудованием требуются специализированные программные средства, обеспечивающие удобство разработки, отладки и внедрения алгоритмов управления.

**OWEN Logic** — это специализированная среда программирования, созданная для разработки алгоритмов управления устройствами компании «ОВЕН». Программа предоставляет графический язык программирования FBD (Function Block Diagram), который соответствует международному стандарту МЭК 61131-3. OWEN Logic поддерживает широкий спектр функциональных возможностей, включая симуляцию работы алгоритмов, взаимодействие с внешними системами через протоколы Modbus и другие инструменты для удобного и быстрого проектирования решений.

Документация к OWEN Logic, опубликованная на официальном сайте компании «ОВЕН» [https://owen.ru/documents], подробно описывает её функциональные возможности, включая работу с программируемыми реле серий ПР100, ПР200 и ПР205, а также с панелями оператора.

Эффективная работа подобных инструментов требует постоянного развития и усовершенствования, чтобы соответствовать новым требованиям пользователей и технологическим вызовам. Это делает актуальной задачу улучшения функционала среды OWEN Logic, добавления новых возможностей и повышения удобства её использования для автоматизации процессов.

Целью данного дипломного проекта является разработка и усовершенствование функционала среды программирования OWEN Logic, а также анализ её возможностей для решения задач автоматизации технологических процессов.

Для достижения этой цели в рамках работы решаются следующие задачи:

1. Изучение архитектуры и существующего функционала OWEN Logic.
2. Анализ пользовательских требований и потребностей в новых функциях.
3. Разработка нового функционала или усовершенствование существующих инструментов.
4. Внедрение изменений в код программы и тестирование их на реальных примерах.
5. Оценка эффективности разработанных улучшений.

В результате реализации проекта будет усовершенствован функционал OWEN Logic, что обеспечит разработчикам и пользователям среды ещё больше возможностей для создания гибких и надёжных решений автоматизации. Разработанное программное обеспечение поможет ускорить процесс проектирования, повысить удобство использования и удовлетворить актуальные потребности специалистов в области автоматизации.

1 Подходы к построению информационных систем

Архитектура программного обеспечения играет ключевую роль в разработке и функционировании информационных систем. В случае с **OWEN Logic** используется подход **Domain-Driven Design (DDD)**, который ориентирован на создание гибких и масштабируемых систем, где бизнес-логика и предметная область играют главную роль в проектировании.

1.1 Принципы DDD

**Domain-Driven Design (DDD)** — это подход к проектированию программных систем, который ставит в центр внимания предметную область и бизнес-логику. В рамках **OWEN Logic** DDD используется для упрощения разработки и улучшения взаимодействия между различными частями системы, а также для более гибкого изменения и расширения функционала в будущем.

Основные принципы DDD, которые применяются в системе, включают:

* **Моделирование предметной области**: Всё программное обеспечение строится с учётом требований и особенностей бизнес-логики, связанной с автоматизацией технологических процессов. Все компоненты системы проектируются в тесном взаимодействии с экспертами в предметной области, чтобы учесть все аспекты работы с программируемыми реле и панелями компании «ОВЕН».
* **Единый язык (Ubiquitous Language)**: В проектировании системы важно использовать общий язык, понятный как техническим специалистам, так и экспертам в области автоматизации. Это означает, что терминология и концепции, используемые в коде, документации и при общении с пользователями, должны быть одинаковыми, чтобы избежать недоразумений и упростить взаимодействие между всеми участниками разработки.
* **Разделение на контексты (Bounded Contexts)**: Система разделяется на различные **контексты**, каждый из которых охватывает определённую часть бизнес-логики. Это позволяет избежать излишней сложности и обеспечивать независимость различных компонентов. Например, контекст для работы с алгоритмами управления может быть отделён от контекста работы с базами данных или внешними системами. Это помогает лучше управлять зависимостями и облегчает поддержку и расширение системы.

1.2 Структура и компоненты на основе DDD

В **OWEN Logic** можно выделить несколько ключевых компонентов, организованных с использованием принципов DDD:

* **Слой домена (Domain Layer)**: Этот слой включает в себя бизнес-логику и модели, которые отвечают за работу с программируемыми реле и панелями. В нем определяются основные сущности, такие как алгоритмы управления, параметры устройств, сценарии работы и другие элементы, важные для автоматизации процессов. Основной задачей этого слоя является обработка данных и выполнение логики, которая не зависит от внешних технологий.
* **Слой приложения (Application Layer)**: Этот слой взаимодействует с пользователем или другими приложениями, предоставляя интерфейсы для работы с бизнес-логикой. В нем находятся сервисы, которые управляют запросами, получают данные и передают их в слой домена
* **Инфраструктурный слой (Infrastructure Layer)**: Этот слой отвечает за взаимодействие с внешними системами и технологическими компонентами, такими как базы данных, внешние устройства, системы мониторинга и т.д.
* **Интерфейсы и внешние компоненты (Interfaces and External Components)**: Это взаимодействие с пользователем, другими приложениями или внешними сервисами, например, через графический интерфейс или командную строку.

Схематическое представление архитектуры DDD, с выделением слоев и контекстов, представлено на рисунке 2.1.

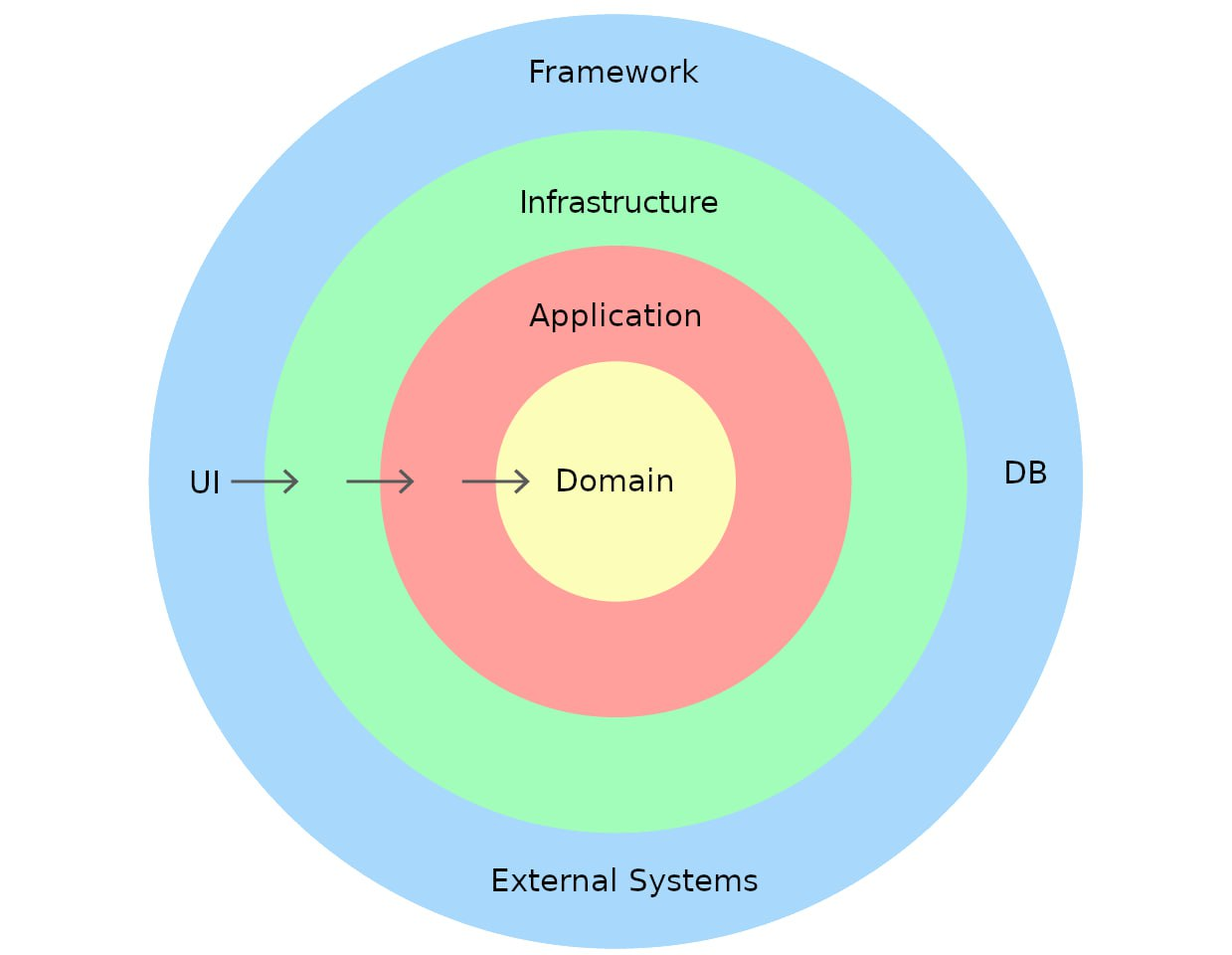


Рисунок 2.1 – Схематическое представление архитектуры DDD, с выделением слоев и контекстов

1.3 Преимущества применения DDD

Использование **DDD** в **OWEN Logic** позволяет достичь нескольких ключевых целей:

* **Модульность и независимость компонентов**: Разделение системы на контексты и использование чётких интерфейсов между ними позволяет создавать независимые компоненты, которые можно легко изменять, заменять или масштабировать.
* **Гибкость и масштабируемость**: Архитектура, основанная на DDD, позволяет гибко адаптировать систему под изменяющиеся требования. Это особенно важно для среды программирования, которая должна поддерживать разнообразие задач, связанных с управлением технологическими процессами.
* **Упрощение взаимодействия между разработчиками и бизнес-экспертами**: Единый язык, используемый в DDD, способствует лучшему пониманию предметной области и уменьшению недопонимания между участниками разработки. Это позволяет быстрее разрабатывать и внедрять новые функции и решать возникающие проблемы.

1.4 Применение DDD для расширяемости

С помощью **DDD** можно легко добавлять новый функционал и расширять программу. Это может быть особенно полезно в будущем, когда появятся новые типы устройств или технологии, которые потребуют интеграции в существующую систему. Благодаря ясному разделению на слои и контексты, расширение системы не будет нарушать её стабильность, а новые функции можно будет добавить с минимальными затратами.

**2 Сравнительный анализ с системами-аналогами**

**2.1 Codesys**

Codesys является одной из ведущих платформ для разработки программного обеспечения для программируемых логических контроллеров (ПЛК). Она поддерживает широкий спектр оборудования от различных производителей и предоставляет многоязыковую среду разработки, соответствующую стандарту IEC 61131-3.

Ключевой функционал:

* Поддержка языков программирования FBD, LD, ST, IL и SFC.
* Мощные средства отладки, включая пошаговое выполнение и точечные остановки (breakpoints).
* Расширенная работа с коммуникациями: поддержка различных протоколов (Modbus, CANopen, BACnet и т.д.).
* Интеграция SCADA-систем для визуализации и мониторинга процессов.
* Широкая библиотека готовых функциональных блоков.
* Возможность управления системой с мобильных устройств.

Недостатки:

* Высокая сложность освоения для начинающих пользователей.
* Платный функционал для ряда возможностей, включая расширенные инструменты диагностики.
* Отсутствие оптимизации под оборудование компании «ОВЕН».

Codesys подходит для работы с многозадачными проектами и интеграции с разными производителями, но требует больше времени и ресурсов на настройку и изучение.

**2.2 TIA Portal (Siemens)**

TIA Portal — это комплексная среда разработки от Siemens, предназначенная для программирования и конфигурирования устройств из линейки SIMATIC, включая ПЛК, HMI и SCADA.

Ключевой функционал:

* Единая среда разработки для всего оборудования Siemens.
* Поддержка языков стандарта IEC 61131-3, включая FBD, LD и ST.
* Визуализация процессов с помощью SCADA-систем.
* Расширенные возможности анализа производительности оборудования.
* Интеграция с системами управления энергопотреблением.

Недостатки:

* Высокая стоимость лицензий.
* Ограниченная совместимость с оборудованием других производителей.
* Отсутствие встроенных симуляторов для быстрой отладки без подключения устройства.

TIA Portal является мощным инструментом для крупных предприятий, использующих оборудование Siemens, однако избыточна для более узких задач автоматизации.

**2.3 Trace Mode**

Trace Mode — это интегрированная система для автоматизации управления, мониторинга и сбора данных (SCADA/HMI), разработанная российской компанией AdAstrA.

Ключевой функционал:

* Создание человеко-машинных интерфейсов (HMI) и SCADA-систем.
* Поддержка большого количества протоколов связи, включая Modbus, OPC и BACnet.
* Интеграция с внешними базами данных для хранения параметров и результатов работы систем.
* Расширенные функции визуализации, включая 3D-графику.
* Инструменты аналитики для оценки эффективности работы системы.

Недостатки:

* Высокие системные требования.
* Сложность настройки для пользователей, не знакомых с системами SCADA.
* Отсутствие встроенной оптимизации для работы с ПР компании «ОВЕН».

Данное решение ориентировано на создание SCADA-систем, а не на программирование логики работы реле и панелей.